NEU_MITLL @ TRECVid 2015: Multimedia Event Detection by Pre-trained CNN Models

Joseph P. Robinson[1], Edward Scott[1], Kevin Brady[3], Charlie K. Dagli[3], and Yun Fu[1,2]

[1]College of Electrical and Computer Engineering, Northeastern University, Boston, Massachusetts
[2]College of Computer and Information Science (Affiliated), Northeastern University, Boston, Massachusetts
[3]Human Language Technology Group, MIT Lincoln Laboratory, Lexington, Massachusetts

ABSTRACT

We introduce a framework for multimedia event detection (MED), which was developed for TRECVID 2015 using convolutional neural networks (CNNs) to detect complex events via deterministic models trained on video frame data. We used several well-known CNN models designed to detect objects, scenes, and a combination of both (i.e., Hybrid-CNN). We also experimented with features from different networks fused together in different ways. The best score achieved was by fusing objects and scene detections at the feature-level (i.e., early fusion), resulting in a mean average precision (MAP) of 16.02%. Results showed that our framework is capable of detecting various complex events in videos when there are only a few instances of each within a large video search pool.

1 Introduction

This report summarizes the performance of a system designed jointly by Synergistic Media Learning (SMILE) Lab of Northeastern University (NEU) and Massachusetts Institute of Technology Lincoln Laboratory (MIT-LL) for the TRECVid 2015 MED task. Specifically, we present results for the 10Ex pre-specified evaluation protocol, which involves 10 positive training videos for 20 events (E21-E40). More information about the MED15 evaluation plan is provided in Section 2 and [2].

Our system uses pre-trained, deeply learned convolutional neural networks (CNNs) as *off-the-shelf* feature extractors, using outputs from either the last or second-to-last fully connected layer as feature vectors to detect complex events in video data. The networks used were VGG-16 [4], Places205-Alexnet, and Hybrid-CNN [6], all of which are further discussed in Section 3.1.

The rest of the paper is organized as follows. First we briefly describe the MED15 task and provided data in Section 2. Then, in Section 3, we introduce our system by discussing the feature extraction work-flow for both training and test phases. In Section 4, experimental settings for each run, along with the performance metric, and computational resources utilized are covered. In Section 5, we present the results, both the overall MAP and event-specific scores for each run. We then conclude in Section 6.

2 MED Task

The goal of MED is to determine whether a particular video contains a particular complex event, such as a *bike trick* or *rock climbing*. As defined in the provided event kits, events typically coincide with the presence of certain objects in certain settings (i.e., scenes) [see Section 2.1.1].

2.1 MED data

2.1.1 Event Kits

An event kit is provided for each event-type. Specifically, each event kit comes with a text description that contains the event name, event definition, event explication, evidential description, and a set of training exemplar videos for

This work is sponsored by the Assistant Secretary of Defense for Research & Engineering under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

the given event.

The MED15 evaluation spans 30 event-types: 20 pre-specified events and 10 Ad-Hoc events. The results presented in this paper are for the pre-specified event classes, which are listed in Table 1.

Table 1. A list of MED15 Pre-Specified Event Types. E021-E030 and E031-E040 come from the MED12 and MED13 video collection, respectfully.

Pre-specified Events

E021	Bike trick	E031	Beekeeping
E022	Cleaning an appliance	E032	Wedding shower
E023	Dog show	E033	Non motorized vehicle repair
E024	Giving Directions	E034	Fixing a musical instrument
E025	Marriage Proposal	E035	Horse riding competition
E026	Renovating a home	E036	Felling a tree
E027	Rock climbing	E037	Parking a vehicle
E028	Town hall meeting	E038	Playing fetch
E029	Winning race w/out a vehicle	E039	Tailgating
E030	Working on metal crafts project	E040	Tuning a musical instrument

Video exemplars are selected to be indicative of a particular event. However, due to the complexity of these high-level event classes, all intra-class variations are unlikely to be represented in the training set. For MED15, there were 3 sets of event kits supported for the evaluation.

1. **0Ex:** No example video clips per event kit.
2. **10Ex:** 10 positive and up to 5 miss/ non-positive clips per event kit.
3. **100Ex:** 100 positive and up to 50 miss/ non-positive clips per event kit.

All experiments reported here used the 10Ex event kits.

2.1.2 Test Search Video Collection

MED15 participants were given a large corpus of videos for the search pool. This corpus, referred to as the Progress Search Set, supports ”blind” testing and, hence, no ground truth was provided.[1]

There are two sets of pre-specified video corpora for teams to select from.

1. **MED15EvalFull:** Approximately 200,000 videos.
2. **MED15EvalSub:** A subset of 32,000 videos from MED15EvalFull.

All tests reported here were with the MED15EvalSub collection.

3 System Overview

As shown in Figure 1, our system uses pre-trained CNN models as *off-the-shelf* feature extractors. These networks accept images as inputs, so each video was first sampled at a rate of about one frame per second [see Figure **??**(a)]. Next, image entropy– a statistical measure of randomness that can be utilized to characterize the texture of an image– of each frame is calculated, which must surpass a minimum threshold value of 0.3. Frames with entropies less than this value are dropped from the sample set, as a small entropy value typically indicates a near perfectly flat color image-frame, either with or without text overlaid (e.g., blank image, credits, logos, etc.). Such frames are useless for

[1] Teams submitted results to NIST for scoring, which consisted of ranked lists of videos, processing speeds, and bodies of text that describe both the evaluation conditions and system descriptions on a run-by-run basis. For each run, events were processed independently and event-specific results were returned by NIST thereafter.

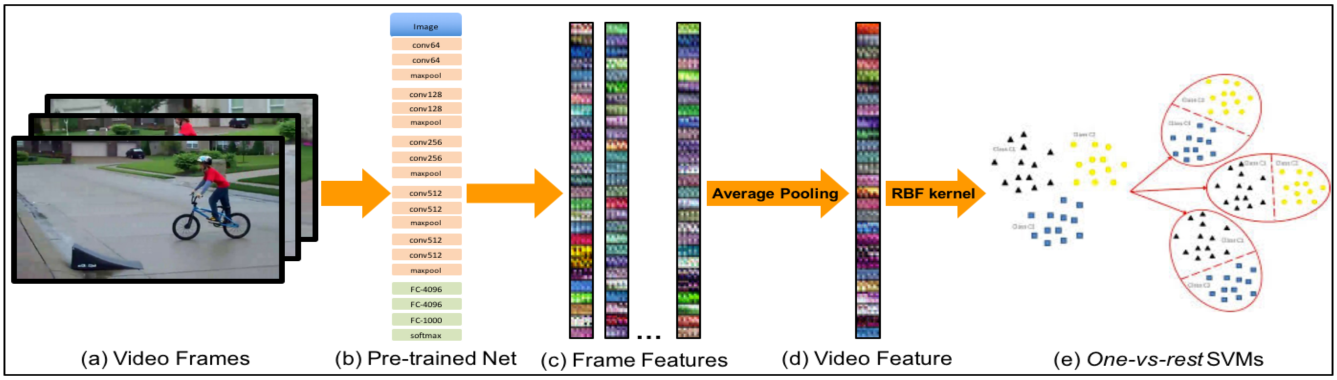

Figure 1. The work flow of our MED system. Given an input video, it is first sampled at approximately one fps (a). Sample frames are then fed forward through pre-trained CNN models (b), yielding a set of frame-level feature vectors (c). Next, a video-level descriptor is formed by average pooling the frame-level features (d). Video descriptor is projected to a non-linear space via RBF kernel, and then passed to the *one-vs-rest* SVM models (e).

our vision-based system and may add unnecessary noise to the training data. A maximum number of 160 frames per video is processed– if a video's frame count exceeds 160, then sample frames are selected at random. Deep features were extracted for each video by passing its sample-frames through the given pre-trained net [see Figure **??**(b)]. Then, a single video-level feature was obtained by average pooling the frame-level features [see Figure **??**(c)]. The following section covers the pre-trained deep nets used in more detail.

3.1 Deep Features

Three pre-trained CNNs were used– VGG-16, Places205-Alexnet, and Hybrid-CNN– which are described as follows:

VGG-16 [4] was trained on the ImageNet ILSVRC 2012 dataset, which consists of 1.3 million images for 1,000 object types [3].

Places205-Alexnet was trained on MIT's Places Database, which consists of 2.5 million images for 205 categories of Places [6]. The architecture imitates the Caffe reference network [1].

Hybrid-CNN was trained on Places Database (205 scene types) and a subset of ImageNet ILSVRC (978 object types) which, in total, consists of about 3.6 million images across 1183 class types [6]. The architecture imitates the Caffe reference network [1].

3.2 Model

Events were modeled using Support Vector Machines (SVM). Typically, SVMs act as binary classifiers with an objective to maximize the margin separating two classes. Following [5], we trained a *one-vs-rest* SVM model for each of the N event-types, i.e., each SVM was trained using the respective in-class data as the positive samples, while all out-of-class data was used as the negative samples.

These event-specific SVM classifiers were based on video-level descriptors applying averaging pooling across the frame-level feature vectors of a given video. Video-level descriptors were projected to a non-linear kernel space via a χ^2 (RBF) kernel, from which linear SVMs were trained.[2] SVM models were stored as the event metadata store, and the features of the test (search) videos were stored as the search-pool metadata. To search for a particular event, videos were ranked according to SVM scores.

[2] Kernel and SVM classifiers were implemented with VLFeat toolbox [5].

4 Experimentation

In this section, we report our results for the TRECVid's MED15 10Ex, PS subtask. We present a total of six runs: one of which was our official submission, while the other five were processed post-evaluation. Runs are summarized in Table 2.

4.1 Performance metrics

According to [2], performance was measured via mean average precision (MAP). For Q events, MAP was determined as follows:

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q} AP(q),$$

AP, as a function of event q, is defined as

$$AP(q) = \frac{1}{P_Q}\sum_{tp=1}^{P_Q} Prec(tp) = \frac{1}{P_Q}\sum_{tp=1}^{P_Q} \frac{tp}{rank(tp)},$$

where P_Q is the number of training exemplars for event Q, which remains constant at 10 throughout each of our runs.

4.2 Computational resources

Development and most processing was done on a local machine equipped with an Intel Core i7-5930K CPU @ 3.50 GHz with 32 GB of memory and a 4GB GEForce GTX 970 GPU running on Ubuntu 14.04 LTS.

5 Results

As listed in Table 3, our official submission, (*Run-1*) returned a MAP of 11.80%. For this, separate SVMs were trained on 1000D VGG-16 outputs and 205D Places-205 outputs. The two sets of SVM scores were averaged for each of the test videos (i.e., late fusion). In contrast, by concatenating the outputs of VGG-16 and Places-205 networks at the feature-level (*Run-5*) the MAP increased to 16.02% (i.e., early fusion).

Interestingly, results from only using VGG-16 outputs (*Run-2*) also outscored our official submission (*Run-1*) at a MAP of 13.26%. In contrast, an *off-the-shelf* hybrid model (*Run-4*), performed similarly to *Run-1*. We can draw a few conclusions from this set of results. Object features (VGG-16) appear to be much more discriminative than scene features (Places-205), but scene features do provide useful information to the system. Additionally, features from the VGG-16 and Places-205 fused together– whether by early or late fusion– is more discriminative than the Hybrid-CNN model, showing it is of worth to use the two networks independently.

Table 2. Run IDs and descriptions. Descriptions list the names of the CNNs used, along with the network layers used as outputs (i.e., features) and the dimensions as seen by the SVMs. Feature fusion schemes are also specified as necessary, i.e., *Run-1* and *Run-5* were fused on the score-level and feature-level, respectfully.

Run ID	**Run Descriptions**
Run-1	VGG-16 + Places205, fc8-layers (1,000D + 205D, respectfully); Averaged SVM scores.
Run-2	VGG-16, fc8-layer (1,000D).
Run-3	Places205, fc8-layer (205D).
Run-4	Hybrid-CNN, fc8-layer (1,186D).
Run-5	VGG-16 + Places205, fc8-layers; Concatenated features vectors (1,205D).
Run-6	VGG-16, fc7-layer (4,096D).

Table 3. Submission scores for each event [AP (%)], along with the overall mean (MAP %). Note that our official submission was *Run-1*, while *Run-2* through *Run-6*) were submitted and scored post-evaluation.

	Run-1	*Run-2*	*Run-3*	*Run-4*	*Run-5*	*Run-6*
E021	17	**26.8**	5.8	21.4	19.4	17.2
E022	2.5	2.4	2.4	1.4	**4**	2.2
E023	26.9	26	14.1	24.8	33.6	**33.8**
E024	1	1	0.7	0.8	**3.3**	0.4
E025	0.3	0.3	**0.4**	0.3	0.3	0.2
E026	**11.1**	8.8	8.7	5.2	10.5	7
E027	25.4	32.5	18.1	29.7	36.8	**43.1**
E028	11	10	11.1	9.6	**17.4**	14.3
E029	19.2	16.9	22.7	12.9	**28.6**	25.4
E030	8.8	9.4	3.8	5.3	**11.9**	8.3
E031	18.8	**40.6**	8.5	37.2	25.6	23.7
E032	3.6	3	3.5	2	3.6	**4.4**
E033	9.8	10.8	2.6	4.5	**19.6**	6.3
E034	3.7	4.7	2.2	2.2	**7.3**	6.1
E035	23.3	22.5	18.3	24.9	**26**	18.9
E036	3.2	3.3	2.5	3.9	**5.9**	2.8
E037	6.7	7.6	6.4	7.3	**9.3**	5
E038	3.5	2.2	4.6	3.7	**4.9**	2.2
E039	30.5	25.2	25.7	27.8	**39.7**	14.6
E040	9.6	11.2	6.5	8.5	**12.7**	11.5
MAP	11.795	13.26	8.43	11.67	**16.02**	12.37

6 Conclusion

We have presented an overview of a multimedia event detection system utilizing pre-trained CNN models to extract meaningful features from video data. Outputs from these models corresponded to the probability that certain objects or scenes were present in a given video. The VGG-16 model proved to be the most discriminative when used independently, resulting in a MAP of 13.26%. Combining VGG-16 and Places-205 by concatenating their outputs (1000-object and 205-scene vectors, respectively) improved MAP to 16.02%. This system represented a strong baseline and starting point for future work. The training scheme could be improved by using different models for the event-type classifiers and by augmenting the data with different feature types. The incorporation of temporal information and additional modalities such as audio and contextual cues will not only improve performance, but also move the system toward zero-shot detection capability, requiring no training exemplars to detect events in video data.

References

1. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
2. Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Wessel Kraaij, Alan F. Smeaton, Georges Quéenot, and Roeland Ordelman. Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2015*. NIST, USA, 2015.
3. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej

Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015.

4. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

5. A. Vedaldi and B. Fulkerson. VLFeat - an open and portable library of computer vision algorithms. In *ACM International Conference on Multimedia*, 2010.

6. Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 487–495. Curran Associates, Inc., 2014.